In the Specification

Please add the following paragraphs on Page 9, after line 11, to read as follows:

In one embodiment, the Object Table essentially contains information for finding a particular value (e.g. AID=surname, ValueNorm="HARVEY") and for retrieving values (e.g. AID=surname, ValueRaw="Harvey"). This table can therefore be split into two tables: SEARCH and ENTRY.

The Search Table is used to resolve filters in the Search service. It is also used to find values during Compare, Modify and ModifyRDN. The Search table contains one row for each attribute value of each entry. Only the normalized values are stored in this table.

The Entry table is used to return values in Reads and Searches. The Entry table contains one row for each attribute value for each entry. The RAW value is the value exactly as initially supplied when the entry was added or modified.

In a further embodiment, every X.500 attribute has a (internationally defined) syntax. X.500 attribute syntaxes define how each attribute should be treated. In all string syntaxes (e.g. Printable, Numeric etc.) superfluous spaces should be ignored. In some syntaxes the case is not important (e.g. Case Ignore String and Case Ignore List) and so the names "Chris Masters", "Chris MASTERS" and " ChRis MaSTeRS " are considered identical.

In order to do comparisons (e.g. search for a particular value), the syntax rules can be applied to create a normalized form (e.g. "CHRIS MASTERS"). If this normalized form is stored in the database, then any variations in input form are effectively removed, and exact matching can be used (which is necessary when using SQL).

Both the normalized data and "raw" data are stored in the database. The "raw" data is necessary so that users can retrieve the data in exactly the same format as it was originally input.